REPLICACIÓN DESCONECTADA DE BASES DE DATOS DISTRIBUIDAS EN DISPOSITIVOS MÓVILES

Mario José Villamizar Cano, Ricardo Andrés Suarez Chaparro and Andrés Mauricio Jiménez Torres, *Estudiantes, Universidad de Los Andes*

***Resumen*—A medida que los dispositivos de computación móvil son cada vez más populares y tienen mejores capacidades, nuevas y sofisticadas aplicaciones y tecnologías pueden ser ejecutadas en ellos, la replicación desconectada de bases de datos distribuidas en dispositivos móviles es una de las tecnologías que pueden ser implementadas sobre estos dispositivos permitiendo que los usuarios puedan realizar operaciones sobre la información replicada en los dispositivos de manera local incluso en modo desconectado. Esta tecnología proporciona nuevas oportunidades en los procesos de negocios de algunas empresas como es el caso de los sistemas de ventas móviles y también en otros contextos como son los ambientes de investigación cooperativos. Sin embargo las nuevas oportunidades brindadas por esta tecnología tienen un costo relacionado con la complejidad en el diseño de los sistemas que la implementan. En este artículo se analizan algunos de los diferentes aspectos o factores que deben ser tenidos en cuenta al diseñar sistemas que utilicen esta tecnología, así como las diferentes técnicas, esquemas o mecanismos que pueden ser utilizados durante el diseño de cada uno de los factores para solucionar los diferentes problemas que se deben enfrentar.**

***Indice de Términos*— Bases de datos, Dispositivos móviles, Operación desconectada, Replicación optimista, Sistemas Distribuidos**

I. INTRODUCCIÓN

LA replicación de datos es una tecnología muy utilizada en los sistemas distribuidos modernos porque permite mejorar la disponibilidad y el rendimiento en el acceso concurrente a los datos realizado por diferentes aplicaciones de un sistema y además permite mejorar algunos procesos de negocio de las organizaciones. Junto con la popularidad y capacidad que han ido adquiriendo los dispositivos de computación móvil como Palms, teléfonos celulares y los computadores portátiles, hoy en día es posible utilizar técnicas de replicación de datos sobre éstos dispositivos haciendo que la tecnología denominada sistemas de bases de datos móviles sea cada día más popular. Una característica importante de estos sistemas de bases de datos es su habilidad para operar de manera desconectada. La desconexión se refiere a la condición en la que un dispositivo móvil no es capaz de establecer comunicación con todos, alguno o ninguno de sus pares. En esa situación un dispositivo no es capaz de accesar a datos compartidos con sus demás pares, lo que podría detener una operación cooperativa que requiera acceso a los datos compartidos entre varios pares, sin embargo, si se ubican o colocan réplicas de datos locales en cada dispositivo se puede permitir que un usuario pueda operar sobre dichos datos de manera desconectada. Esta capacidad permite proporcionar nuevas oportunidades en los procesos de negocios de las empresas y/o un mejoramiento en las opciones o características ofrecidas por aplicaciones o procesos ejecutados en otros contextos.

Un ejemplo en el cual la operación desconectada permite mejorar las oportunidades de los negocios es lo que se conoce como sistemas de ventas móviles, estos sistemas le permiten a los vendedores de una empresa realizar su proceso de ventas de una forma mucho más ágil, eficiente y productiva, ya que permiten que el vendedor lleve información importante de los clientes, inventarios, productos, entre otros, en los dispositivos móviles, los cuales en este caso generalmente son computadores portátiles. Un aspecto importante de la nueva forma de realizar el proceso de ventas es que ahora los dispositivos de los vendedores no requieren tener conectividad permanente con otros dispositivitos o sitios principales, lo cual permite un ahorro en los costos que se pueden incurrir al utilizar esquemas de conectividad permanente con otros sitios, también permite que este proceso pueda ser realizado en zonas alejadas donde no hay enlaces de comunicación. Otro beneficio que se obtiene al utilizar esta tecnología, es que el vendedor le puede ofrecer a los clientes una atención más personalizada con información más detallada de los productos, adicionalmente la tecnología permite agilizar el proceso de despacho de los pedidos realizado en la sede principal de la empresa, ya que se puede realizar el envío de una forma más rápida a través de medios como Internet, lo cual conlleva a automatizar en gran parte el despacho de los pedidos, pues estos despachos ahora se pueden realizar de una forma más ágil, rápida y con pocos errores. Otro ejemplo en el que la operación desconectada puede ser utilizada es en el trabajo colaborativo realizado por un grupo de científicos, en este caso lo más importante en que los científicos necesitan compartir

información pero la manipulación de dicha información debe ser realizada de forma local en el sitio o dispositivo de cada científico, una especificación más detallada de este ejemplo es el caso de un científico que necesita involucrarse en un ambiente donde no existen enlaces de comunicación por un largo periodo de tiempo (días, semanas, meses) como es el caso de una selva o desierto, dicho científico durante su estadía en el ambiente requiere analizar información anteriormente recopilada por otros colegas que le permita tomar decisiones en las acciones a realizar, además el científico también requiere poder registrar nueva información recopilada del ambiente de trabajo que posteriormente debe ser enviada al resto de los integrantes del grupo para poder realizar análisis más exhaustivos sobre la nueva información.

Con base en los ejemplos se pueden comprender las nuevas oportunidades que brinda el uso de la "replicación desconectada de bases de datos distribuidas en dispositivos móviles", es por eso que en este artículo se analizan y especifican de manera general algunos de los aspectos principales que deben ser tenidos en cuenta al diseñar sistemas que utilicen esta tecnología, ya que a pesar de las nuevas oportunidades que brinda, el uso de esta tecnología lleva a los diseñadores de sistemas a enfrentarse a una serie de problemas que en ocasiones requieren el planteamiento de nuevas y complejas soluciones.

El artículo está organizado de la siguiente forma: en la sección 2 se hace un breve descripción de algunos de los trabajos relacionados que han sido desarrollados en otras comunidades académicas, en la sección 3 se realiza una especificación de los diferentes conceptos que deben ser tenidos en cuenta al trabajar con esta tecnología así como las ventajas y desventajas de implementarla. En la sección 4 se describen y analizan algunos de los diferentes aspectos o factores que deben ser tenidos en cuenta al diseñar sistemas que utilicen esta tecnología, así como las diferentes técnicas, esquemas o mecanismos que pueden ser utilizados durante el diseño de cada uno de los factores para solucionar los diferentes problemas que se deben enfrentar. Finalmente en la sección 5 se establecen las conclusiones obtenidas con la elaboración del artículo..

II. Trabajos relacionados

Actualmente se tiene referencia de una gran cantidad de trabajos relacionados que abarcan parcialmente el tema tratado en el artículo. En el artículo [1] se proporciona una introducción sobre el funcionamiento de los esquemas de replicación de bases de datos en dispositivos móviles, el artículo [2] se enfoca en realizar un análisis de la forma como debe ser realizado el proceso de reconciliación de transacciones en ambientes de replicación móvil que permiten la actualización de réplicas de forma desconectada y funcionan bajo el modelo cliente/servidor, en [3] se propone un protocolo para garantizar la consistencia de los datos replicados en ambientes de computación móvil, en [4], [6] y [7] se analizan y diseñan protocolos y algoritmos que permiten emplear esquemas dinámicos de replicación utilizando bases de datos móviles, los cuales buscan mejorar el rendimiento y el acceso a los datos estableciendo sistemas dinámicos de replicación que asignan las replicas de datos a los dispositivos móviles teniendo en cuenta los datos usados y la ubicación de los usuarios que utilizan dichos dispositivos. En [5] se propone un modelo para garantizar la escalabilidad de los sistemas de bases de datos móviles con conectividad intermitente. En [8] se analizan algunos de los requerimientos o factores que deben ser tenidos en cuenta al utilizar técnicas de replicación de datos en ambientes móviles, en [9] se analizan los diferentes aspectos que deben analizarse cuando se diseñan sistemas que utilizan técnicas optimistas de replicación. En [10] se hace una descripción del sistema Roam diseñado para garantizar la escalabilidad de un esquema de replicación de datos específico, en este sistema se utiliza un esquema de propagación de operaciones utilizando como referencia los esquemas de replicación de los ambientes P2P. En [11] se especifican los diferentes modos de operación desconectada que puede tenerse en un esquema de replicación de datos en ambientes móviles y en el artículo [12] los autores se enfocan en el desafío de mantener las restricciones de integridad de las bases de datos relacionales en presencia de desconexiones frecuentes de los sitios que realizan las actualizaciones, con el fin de garantizar la autonomía en el procesamiento de transacciones llevado a cabo en cada uno de los dispositivos móviles.

III. Conceptos y factores que involucra el uso de la tecnología

En esta sección se describen en detalle las limitaciones que enfrentan las técnicas de replicación tradicionales basadas en los esquemas de replicación pesimistas, también se definen las ventajas y desventajas de los esquemas de replicación optimista, se especifican los problemas de consistencia y accesos concurrente que deben enfrentarse al utilizar la tecnología de replicación desconectada y finalmente se realiza la definición de los diferentes conceptos que serán utilizados a los largo del artículo.

A. La replicación pesimista y sus limitaciones

Las técnicas de replicación tradicionales tienen como meta mantener una única copia de datos consistente, la cual da a los usuarios la ilusión de tener una sola copia de los datos que proporciona un alta disponibilidad [9]. La meta de las técnicas de replicación tradicionales puede ser lograda siguiendo muchos caminos, sin embargo todos los caminos convergen a una decisión básica común: bloquear el acceso a todas las réplicas de un mismo objeto mientras se realiza una actualización sobre una de las réplicas, una vez que la actualización es realizada el cambio es propagado por el sitio que lo realizó a todos los demás sitios y todas las réplicas son desbloqueadas. Debido a que solo un sitio puede actualizar

una réplica de un objeto en un mismo instante de tiempo estas técnicas son denominadas “pesimistas”. El uso de estas técnicas involucra un rendimiento adecuado cuando se emplean en ambientes totalmente controlados como es el caso de las redes de área local en los cuales la latencia de las comunicaciones en muy baja y las fallas en los enlaces de comunicación son poco comunes, sin embargo al tratar de utilizar estas técnicas en ambientes como las redes de área amplia o Internet, no se puede esperar de estas técnicas un buen rendimiento y disponibilidad por tres razones principales [9].

Primero, la Internet mantiene comunicaciones poco confiables y adicionalmente los dispositivos de computación móvil que requieren conectividad intermitente o periódica son cada vez más populares, esto hace que si se utilizan una técnica de replicación pesimista en estos ambientes, al realizar el proceso de sincronización entres los diferentes sitios y utilizar un algoritmo distribuido para bloquear todas las réplicas de un mismo objeto, si uno de los sitios no está disponible o su enlace de comunicación con los demás sitios ha fallado con seguridad el algoritmo quedaría bloqueado indefinidamente.

Segundo, las técnicas de replicación pesimistas tienen poca escalabilidad en redes de área amplia ya que a medida que el número de sitios o dispositivos se incrementa, la disponibilidad y el rendimiento se ven seriamente afectados debido al constante bloqueo (en ocasiones indefinido) de las réplicas de los objetos que ocasionan las operaciones de actualización realizadas sobre los objetos desde los diferentes sitios.

Tercero, algunas personas al realizar sus actividades requieren poder operar sobre datos compartidos de forma autónoma o desconectada, en estos casos es mejor permitir las operaciones de actualización sobre los objetos de manera independiente, evitando el problema de la sincronización en cada actualización característico de las técnicas de replicación pesimistas. El hecho de permitir actualizaciones autónomas conlleva a que en los sistemas de replicación optimista sea necesario establecer políticas de detección y solución de conflictos, ya que en estos sistemas dos replicas de un objeto pueden ser actualizadas concurrentemente desde dos sitios diferentes lo cual ocasionaría conflictos al momento de realizar el proceso de reconciliación de todas las operaciones realizadas sobre las diferentes réplicas de un objeto.

B. Ventajas y desventajas de la replicación desconectada de bases de datos

El uso de esta tecnología puede solucionar los problemas mencionados en el apartado anterior ya que esta tecnología emplea el uso de técnicas optimistas que proporcionan ciertas ventajas pero que también tienen algunas desventajas.

Al implementar esta tecnología se pueden obtener las siguientes ventajas: Primero, un mejoramiento en la disponibilidad de los datos, ya que los diferentes sitios pueden operar sobre las réplicas locales de los datos incluso cuando los enlaces de comunicación fallan u otros sitios no están disponibles. Segundo, es flexible con respecto a la topología de red utilizada ya que en esta tecnología se pueden emplear diferentes técnicas o algoritmos de propagación para propagar las operaciones realizadas sobre cada objeto a todos los sitios, sin tener un conocimiento detallado o incluso desconocido de los esquemas de interconexión utilizados entre los diferentes sitios y/o dispositivos. Tercero, las técnicas de replicación optimistas empleadas en la tecnología permiten mejorar la escalabilidad de todo el sistema debido a que se requiere una baja sincronización entre todos los sitios. Cuarto, permite a los usuarios que requieren el acceso a los datos operar de manera autónoma. Quinto, permite ejecutar transacciones de manera local en cada sitio denominadas “transacciones tentativas” y posteriormente (bajo demanda o periódicamente) son enviadas a lo sitios maestros los cuales se encargan de hacer el “commit global o tentativo” de las transacciones y posteriormente replicar las transacciones a todos los demás sitios.

Sin embargo a pesar de los beneficios que causa el uso de esta tecnología, su uso también implica un costo en la relación que existe entre la disponibilidad y la consistencia de los datos, costos característico de los las tecnologías que emplean técnicas de replicación optimistas. Teniendo en cuenta la naturaleza optimista de esta tecnología los diseñadores que diseñan sistemas que la implementan deben enfrentar los desafíos de las réplicas divergentes y la detección y solución de conflictos causados entre transacciones concurrentes que son realizadas desde diferentes sitios sobre las réplicas de un mismo objeto. Por esto esta tecnología debe ser aplicada en sistemas que permitan la inconsistencia ocasional o eventual de datos y los problemas que involucran los conflictos entre las transacciones, afortunadamente en el mundo hay muchas aplicaciones que toleran este tipo de problemas relacionados con las inconsistencias y los conflictos.

C. El problema de la consistencia y el acceso concurrente

Teniendo en cuenta que el uso de la tecnología de replicación desconectada permite la actualización concurrente de varias réplicas de un mismo objeto, dos aspectos claves que deben ser tenidos en cuenta son la consistencia y el acceso concurrente, pues al contrario de las técnicas “pesimistas” en las cuales al momento de actualizar la réplica de un objeto se realiza un proceso de sincronización que permite actualizar todas las réplica de un mismo objeto dentro de las misma transacción, la naturaleza autónoma de los esquemas de replicación optimista hace que este proceso de sincronización no puede ser realizado. Por lo tanto uno de los desafíos más grandes al utilizar esta tecnología se basa en garantizar la consistencia de los datos, para garantizar la consistencia en estos sistemas se deben tener en cuenta dos criterios [3]. Primero, el DBMS debe asegurar que la ejecución concurrente

de un conjunto de transacciones sobre una base de datos replicada debe ser equivalente a la ejecución serial del mismo conjunto de transacciones sobre una base de datos no replicada, este criterio es denominado serialización de una sola copia. Segundo, el DBMS debe asegurar que los diferentes estados de las réplicas de un mismo objeto deben convergir a un estado final convergente en un instante dato, este criterio es denominado consistencia eventual. La forma como pueden gestionarse la consistencia y el acceso concurrente a las diferentes réplicas es descrita de manera más detallada en secciones posteriores.

D. Conceptos utilizados en la replicación desconectada de bases de datos

Cuando se trabaja con esta tecnología es muy conveniente tener claros los diferentes términos y conceptos utilizados, por eso en esta sección son definidos de forma general los diferentes términos que serán utilizados en las siguientes secciones del artículo.

Réplicas, sitios y dispositivos móviles. Las técnicas de replicación utilizan el término de réplica para referirse a la mínima unidad de replicación, en este caso una réplica hacer referencia a una copia de un conjunto de objetos (tablas, vistas, procedimientos o funciones almacenada, entre otros) de un DBMS que es almacenada en algún lugar, dicho lugar hace referencia a un sitio maestro y/o un sitio esclavo, los dispositivos móviles son denominados sitios esclavos y los servidores estáticos con grandes capacidades de procesamiento son denominado sitios maestros, por ello cuando se menciona que una réplica es almacenada o actualizada en diferentes sitios, se hace referencia a que los sitios involucrados pueden tener el rol de maestros y/o esclavos. A lo largo del artículo los conceptos de réplica y sitio son utilizados intercambiablemente.

Transacción. Una transacción hace referencia a un conjunto de operaciones de lectura y escritura realizadas sobre la réplica de un objeto. En la tecnología analizada las transacciones en este caso son eventualmente ejecutadas sobre la réplica del sitio local donde es efectuada, permitiéndole al usuario continuar trabajando con base en las diferentes transacciones ejecutada de manera local. Cuando el sitio esclavo se sincroniza con un sitio maestro los cambios efectuados sobre las réplicas del sitio esclavo son enviados al sitio maestro quien se encarga de propagar los cambios a los demás sitios esclavos y/o maestros.

Propagación y ordenamiento de transacciones. Los sitios maestros son los encargados de realizar el ordenamiento de las transacciones enviadas por los demás sitios y propagar los cambios a todos los sitios esclavos, este esquema permite que un sitio reciba todas las actualizaciones realizadas por los demás sitios sobre las réplicas de datos que mantiene y adicionalmente evita que los sitios esclavos o los dispositivos móviles tengan que sincronizarse directamente, sin embargo, la forma propagar las transacciones ocasiona que se deba tener un adecuado control en el orden en que deben ser ejecutadas las transacciones, esto se debe a que la propagación de las transacciones realizadas en un sitio son transmitidas un tiempo después de que son ejecutadas localmente, por esto todos los sitios no siempre reciben las transacciones en el mismo orden. Debido a esto, cada sitio debe realizar un ordenamiento apropiado de las transacciones recibidas que han sido ejecutada en otros sitios, esto se puede realizar a través del uso de una política de ordenamiento de transacciones que se encarga de garantizar que la ejecución de las transacciones en todos los sitios va a proporcionar el mismo resultado, por ello en esta tecnología se usa el término de ejecución tentativa de transacciones ya que esta forma de operación permite que un sitio pueda ordenar repetidamente las transacciones enviadas por los demás sitios garantizando que en algún momento el orden de las transacciones ejecutada en todos los sitios debe ser el mismo.

Detección y solución de conflictos. Como se ha mencionado anteriormente en esta tecnología se puede presentar la situación en que múltiples sitios puedan actualizar concurrentemente varias réplicas de un mismo objeto, lo que ocasiona conflictos al momento de realizar el proceso de reconciliación de transacciones. Una política que se puede establecer es que si ocurre un conflicto entre dos transacciones, se elimine la transacción más antigua, sin embargo esto ocasionaría pérdida de actualizaciones y en muchos sistemas estas pérdidas no son tolerables. Una mejor solución para solucionar el problema se basa en establecer políticas que permitan detectar cuando dos o más transacciones están en conflicto y una vez detectado un conflicto se procedan a utilizar una serie de políticas de resolución de conflictos para solucionarlo.

IV. A.Opciones de diseño en la replicación desconectada de bases de datos en los dispositivos móviles

El objetivo más importante que debe ser tenido en cuenta en la tecnología analizada se centra en garantizar la consistencia de los datos, para ello los diseñadores de sistema que implementan esta tecnología deben tener en cuenta diferentes aspectos en el diseño que pueden influir notablemente en el logro del objetivo principal, es por ello que en esta sección se analizan los principales aspectos de diseño que deben ser tenidos en cuenta al utilizar la tecnología.

A. Número de sitios maestros

Como se mencionó anteriormente cada sitio en el sistema puede cumplir el rol de maestro o el rol de esclavo (si se diseñara el sistema con un enfoque de ambiente P2P podría cumplir ambos roles). Un sitio maestro es aquel con el cual se sincronizan periódicamente o por demanda los sitios esclavos con el fin de intercambiar las diferentes actualizaciones efectuadas mientras los sitios esclavos estuvieron desconectados, esto por lo tanto ubica a un sitio maestro como

el punto central o concentrador utilizado por los esclavos para intercambiar actualizaciones. Generalmente los sitios maestros son ubicados en servidores estáticos con grandes capacidades de almacenamiento y procesamiento los cuales esperan por las solicitudes de actualización de los sitios esclavos, por el contrario los sitios esclavos hacen referencia a los dispositivos móviles que contienen réplicas de los datos almacenados en el sitio maestro.

Uno de los principales aspectos que debe ser analizado es el número de sitios maestros que se requieren en el sistema. Los sistemas que tienen un solo sitio maestro denominados single-master son simples pero pueden tener problemas de escalabilidad, sobrecarga y lentitud en la propagación de las actualizaciones cuando el número de sitios esclavos aumenta considerablemente, además de que el hecho de tener un solo sitio maestro representa un punto único de falla para todo el sistema, puesto que si el sitio maestro se cae las actualizaciones realizadas en los sitios esclavos no pueden ser intercambiadas.

Los sistemas de varios sitios maestros denominado multi-master permiten que el control de las actualizaciones y los procesos de sincronización con los sitios esclavos puedan ser efectuados desde diferentes sitios, lo cual le permite al sistema tener una mayor escalabilidad ya que se puede balancear la carga ocasionada por los procesos de sincronización con lo sitios esclavos y una mayor disponibilidad ya que se evita que exista un punto único de falla que haga que todo el sistema deje de funcionar, sin embargo, estos sistemas son significativamente más complejos, especialmente en lo que respecta a garantizar la consistencia de los datos ya que se deben buscar soluciones más complejas a los problemas relacionados con el ordenamiento de las transacciones y la políticas de detección y solución de conflictos.

B. Modo de operación desconectada

Al trabajar con esquemas de replicación desconectada en dispositivos móviles uno de los aspectos que debe ser tenido en cuenta es el modo de operación desconectada que se requiere en los sitios, dichos modos están basados en los permisos de lectura y escritura asignados a las réplicas, estos permisos determinan las operaciones que pueden ser realizadas sobre las réplicas almacenadas en los diferentes sitios. En esta sección se analizan los principales permisos que pueden tener los sitios maestros y esclavos cuando estos últimos operan de manera desconectada. Las siglas utilizadas DB y I (Si) hacen referencia a toda la base de datos y a una réplica de la base de datos respectivamente.

Modo desconectado de solo lectura. En este modo un sitio esclavo decide desconectarse del sistema. Las réplicas de la base de datos del sitio desconectado llegan a ser de lectura únicamente mientras el sitio se encuentra en dicha situación, en el servidor central se permite acceso de lectura y escritura en toda la base de datos. Este modo es ilustrado en la figura 1 (las líneas continuas indican permiso de acceso de escritura y lectura, y las líneas punteadas indican permiso de solo lectura).

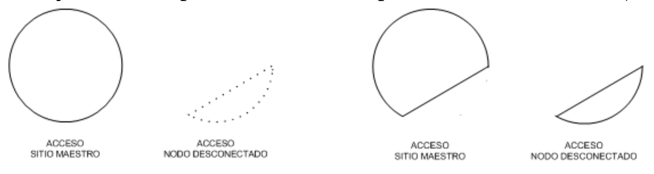


Figura 1. Modo desconectado de solo lectura

Figura 2. Modo desconectado de escritura – Operación pesimista - Partición DB

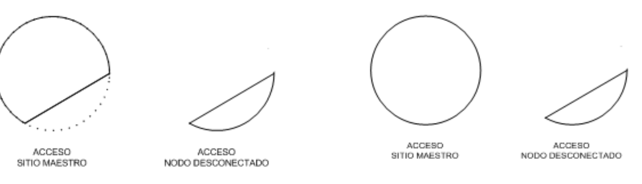


Figura 3. Modo desconectado de escritura – Operación pesimista - Con sistema de lectura

Figura 4. Modo desconectado de escritura – Operación optimista

Modo desconectado de escritura – Operación pesimista. En este modo cuando un sitio esclavo se desconecta y comienza a trabajar de forma desconectada el sitio solo es capaz de realizar actualizaciones sobre un porción de los datos denominada I (Si), en este modo de replicación se pueden encontrar dos variaciones de este modelo que determinan el tipo de acceso a los datos que es permitido. La primera variación es denominada Partición DB (partición de la base de datos), en este modo la base de datos es dividida en dos partes denominadas I (Si) y DB - I (Si). El sitio desconectado es el único que tiene acceso exclusivo de lectura/escritura a I (Si) y nadie más tiene acceso a su porción de datos, de la misma manera el sitio maestro es el único que tiene acceso de lectura/escritura al resto de los datos, es decir a DB-I (Si). Esto es ilustrado en la figura 2. La segunda variación es denominada Con sistema de lectura, este modo permite a un nodo desconectado tener acceso exclusivo de lectura y escritura sobre una(s) réplica(s) de los datos denominada I (Si) y en el sitio maestro se permite acceso de lectura sobre toda la base de datos y acceso de escritura sobre la porción denotada DB-I (Si). Esto es ilustrado en la figura 3.

Modo desconectado de escritura – Operación optimista. En este modo varios sitios maestros o esclavos comparten los permisos de lectura y escritura sobre una(s) réplica(s) de los datos. Este modo se basa en que las escrituras pueden ser realizadas tanto en el sitio maestro como en los sitios esclavos que operan de forma desconectada, por lo tanto los nodos desconectados pueden realizar actualizaciones concurrentemente sobre varias réplicas de un mismo objeto. Este último modo de desconexión es donde se presentan problemas más complejos en lo que respecta a la seriabilidad de las transacciones y consistencia de los datos, ya que en este modo se permite que un objeto sea actualizado por varios sitios al mismo tiempo, lo cual puede hacer que se presenten conflictos, esto hace que al utilizar este modo se requieran mecanismos que permitan detectar los conflictos así como mecanismos para solucionarlos. Este tema es descrito con más

detalle en secciones posteriores. Todo este artículo se basa en los aspectos de diseño que deben ser tenidos en cuenta para la solución de los problemas en los sistemas que utilizan este modo de desconexión. El artículo se centra en este modo de desconexión ya que es el más utilizado y además porque los demás modos de operación presentan problemas menos complejos.

C. *Definición de la topología de propagación*

La topología en estrella simple se basa en que todos los sitios esclavos intercambian las actualizaciones realizadas sobre sus réplicas con un único sitio maestro, quien se encarga de propagarlas a todos los demás sitios.

Otra topología, la implementada en sistemas que utilizan una arquitectura de dos niveles, se basa en que en un nivel se ubican lo dispositivos que deben estar fuertemente conectados como es el caso de los sitios maestros, este nivel es denominado nivel de los sitios centrales, en el otro nivel se ubican los sitios que deben estar débilmente conectados como los son los sitios esclavos o dispositivos móviles, este nivel se denomina nivel de sitios móviles. En los sitios centrales se usan a menudo técnicas de replicación pesimista, garantizando la consistencia de los datos entre todos los miembros de este nivel, pero en cambio en los sitios móviles se emplean técnicas de replicación optimista en sus comunicaciones con los sitios maestros. Esta arquitectura de dos niveles proporciona una mejor escalabilidad, disponibilidad y balanceo de carga. Un ejemplo de una arquitectura en dos niveles puede ver en la siguiente figura:

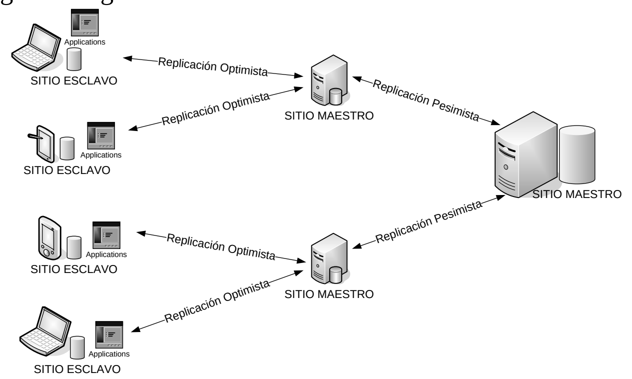


Figura 5. Esquema de una arquitectura de dos niveles

Tomando como referencia sistemas que no se relacionan con las bases de datos pero si con técnicas de replicación como es el caso de Roam [10], el cual es un sistema diseñado para funcionar en un ambiente P2P, se podrían plantear otras topologías como es la organización de los esclavos en anillos, en donde en cada anillo hay un sitio maestro, una característica de la topología de este sistema es que los sitios esclavos pueden intercambiar información directamente sin intervención del sitio maestro del anillo al que pertenecen y adicionalmente el sitio maestro de cada anillo no es un servidor centralizado sino un par establecido mediante un proceso de consenso o elección ejecutado por todos los pares del anillo. En este sistema los sitios maestros son los que se encargan de intercambiar información con sitios maestros de otros anillos, con el fin de que los cambios realizados puedan ser informados a todos los anillos que forman el sistema.

Las tres topologías planteadas son las más comunes en sistemas que utilizan la replicación, sin embargo en los sistemas relacionados específicamente con replicación de bases de datos las dos primeras topologías son las más utilizadas ya que la topología en anillo es poco utilizada pues normalmente la idea de la replicación desconectada de las bases de datos es manejar el esquema en el cual los sitios esclavos realizan operaciones de manera desconectada sobre las replicas de los datos que posteriormente debe ser enviadas a un sitio central para su análisis o procesamiento, por lo tanto el intercambio de información entre esclavos no siempre es útil. Adicionalmente a las topologías planteadas pueden platearse nuevas topologías o topologías mixtas, sin embargo, lo más importante que se debe tener en cuenta al definir la topología a utilizar es buscar un equilibrio entre los diferentes aspectos como son: la escalabilidad del sistema, la velocidad de propagación de las actualizaciones entre sitios, el balanceo de carga en todos los sitios, la disponibilidad y redundancia del sistema, y los requerimientos específicos que se deben cumplir en el sistema. En sistemas donde hay pocos dispositivos móviles una topología en estrella simple es suficiente, sin embargo, cuando el número de dispositivos aumenta una arquitectura de dos niveles puede mejorar el rendimiento general del sistema.

D. *Definición de la forma de propagar las actualizaciones: Transferencia de estado vs. Transferencia de operación*

Durante el proceso de reconciliación de las transacciones un aspecto que debe ser analizado es la forma como se propagan las transacciones entre los sitios maestros y los sitios esclavos. Para ello se puede optar por uno de los dos esquemas que existen que son: los sistemas por transferencia de estado y los sistemas por transferencia de operación. En un sistema en el que se utiliza propagación por transferencia de estado al momento de realizar una actualización sobre un objeto se necesita enviar el nuevo estado completo del objeto, por otra parte es un sistema que utiliza propagación por transferencia de operaciones al momento de propagar la actualizaciones realizadas sobre un objeto se envían las operaciones o transacciones realizadas sobre el objeto.

La transferencia de estados es más simple en su forma de actualizar la actualización porque requiere únicamente del envío del nuevo estado de una réplica a todas las demás réplicas, por otro lado la transferencia de operaciones es un poco más compleja ya que cada sitio debe mantener una historia de las transacciones y el orden en que fueron ejecutadas sobre un objeto determinado. Los sistemas por transferencia de operaciones son más eficientes sobre todo cuando los objetos actualizados son de gran tamaño, por ejemplo un sistema por transferencia de estado puede requerir la transferencia completa de una tabla cada vez que un registro es actualizado, por el contrario en un sistema por transferencia

de operaciones bastaría con enviar la sentencia SQL que realizó la modificación en el registro actualizado, resultando en un reducción inmensa en el tráfico de red generado en la propagación de la actualización y en el tiempo de procesamiento de las actualizaciones al ser recibidas en los diferentes sitios.

En el caso de la replicación desconectada de bases de datos en dispositivos móviles se considera más adecuado el uso de los mecanismos de transferencia de operación ya que éstos sistemas se prestan para permitir el envío de las operaciones realizadas sobre los objetos como son las sentencias SQL, esto permite disminuir considerablemente el tráfico de red generado al momento de realizar el proceso de sincronización entre los dispositivos móviles y los sitios maestros. Adicionalmente los sistemas que utilizan transferencia de operaciones permiten definir políticas de detección y resolución de conflictos mucho más eficientes y flexibles, ya que por ejemplo, si dos sitios esclavos realizan una actualización concurrente sobre dos registros diferentes de una misma tabla, lo más probable si se utiliza propagación por transferencia de estado es que cuando el sitio maestro reciba las actualizaciones enviadas por los esclavos detecte que ambas transacciones están en conflicto ya que el objeto de referencia para actualizar es la tabla completa y en el ejemplo la tabla fue actualizada en el mismo instante de tiempo lo que hace que el sitio maestro tenga que tomar una decisión sobre el conflicto generado. Sin embargo si en el mismo ejemplo se utiliza propagación por transferencia de operación los sitios esclavos enviarían al sitio maestro las sentencias SQL que afectan a dos registros diferentes de la misma tabla, al recibir las dos actualizaciones el sitio maestro no detectaría ningún conflicto ya que en este caso el objeto de referencia es el registro (y no la tabla) lo cual permite que ambas actualizaciones puedan ser ejecutadas exitosamente sin necesidad de versen involucradas en algún conflicto.

E. Ordenamiento de las transacciones: Sintáctico vs. Semántico

Las políticas de ordenamiento de las transacciones tienen como meta ordenar las transacciones en cada uno de los sitios de todo el sistema. Las políticas de ordenamiento pueden ser clasificadas en sintácticas y semánticas. Las políticas sintácticas ordenan las transacciones basadas en la información acerca de cuando, dónde y por quien fueron ejecutadas las transacciones, generalmente las políticas de ordenamiento sintáctico van encaminadas a garantizar la seriabilidad de todas las transacciones ejecutadas en todos los sitios. El ordenamiento basado en marcas de tiempo es uno de los mecanismos más populares utilizado en las políticas de ordenamiento sintáctico. Por otra parte, las políticas de ordenamiento semántico explotan las propiedades semánticas de las transacciones u operaciones realizadas sobre los objetos replicados, como son las propiedades de la conmutación y la idempotencia, estas políticas reducen en gran medida los conflictos y la frecuencia de rollbacks en el sistema. El ordenamiento semántico solo puede ser utilizado en sistemas que utilizan políticas de propagación de transacciones por transferencia de operación ya que por la naturaleza del uso de operaciones en las políticas de ordenamiento semántico, a los sistemas de transferencia de estado no se le pueden aplicar políticas de ordenamiento semántico.

Las políticas sintácticas son más simples pero en los sistemas diseñados pueden aumentar la cantidad de conflictos generados y el número de operaciones innecesarias puede aumentar. Considere el ejemplo en el que dos sitios esclavos actualizan el mismo dato x que es de tipo entero, se hace la suposición que el valor inicial obtenido para x en ambos sitios desde un sitio maestro es x=3 y que el ordenamiento de las operaciones se realiza utilizando una política sintáctica basada en marcas de tiempo, el sitio A actualiza a x efectuando la operación x=x*2 en un tiempo lógico común del sistema t=5, el sitio B actualiza a x efectuando la operación x=x*5 en un tiempo lógico común t=6, se supone que el sitio B inicia el proceso de sincronización primero con el servidor maestro y envía su operación x=x*5, el sitio maestro recibe la actualización y procede a efectuar la operación x=x*5=3*5 asignando el nuevo valor x=15, posteriormente el sitio A envía su operación al sitio maestro, al recibirla este detecta que la operación del sitio A fue realizada primero que la operación anteriormente enviada por el sitio B, lo cual hace que el sitio maestro proceda a: deshacer la operación de actualización recibida del sitio B haciendo x=3, aplicar la operación enviada por A haciendo x=x*2=3*2=6 y ejecutar nuevamente la operación del sitio B haciendo x=x*5=6*5=30. Sin embargo si se analiza se puede comprender que no es necesario que el sitio maestro deshaga la operación, ya que en este caso la propiedad conmutativa de ambas operaciones permite que éstas puedan ser ejecutadas en cualquier orden, si en el ejemplo en el sitio maestro se hubiera utilizado una política de ordenamiento semántico no hubiera sido necesario deshacer la operación que inicialmente se había efectuado.

Los mecanismos utilizados por las políticas sintácticas que pueden ser definidos en el sistema o que ya vienen incluidos en los actuales DBMS se basan principalmente en marcas de tiempo basada en relojes lógicos o también denominado relojes de Lamport o en relojes físicos estableciendo un mecanismo para establecer una hora común aproximada en el sistema o utilizando protocolos ya definidos como es el caso del NTP (Network Time Protocol). Los mecanismos implementados en las políticas de ordenamiento semántico se basan en la explotación de la propiedad conmutativa de las operaciones, el ordenamiento canónico, las transformaciones operacionales y el establecimiento de restricciones entre transacciones. La explicación detallada de los diferentes mecanismos que pueden ser utilizados en las políticas de ordenamiento sintáctico y semántico se sale del objetivo de este artículo, una explicación más detallada de cada uno de los mecanismos puede encontrarse en [9].

Con base en lo anterior se puede ver que el tipo de política de ordenamiento de transacciones a establecer en el diseño de este tipo de sistemas es un aspecto muy importante que debe ser analizado, pues las políticas establecidas pueden influir en el rendimiento del sistema y en el número de conflictos detectados Como se explicó las políticas sintácticas son más simples de implementar, sin embargo en sistemas con alta carga generada por las actualizaciones de los diferentes sitios las políticas de ordenamiento semántico pueden proporcionar un mejor rendimiento en el sistema.

F. Gestión de conflictos

La gestión de conflictos es un factor clave en el diseño de sistemas que utilizan técnicas de replicación optimistas, por esto en la replicación desconectada de bases de datos distribuidas en dispositivos móviles deben emplearse una serie de políticas que se encarguen de gestionar los diferentes conflictos que pueden ocurrir, estas políticas pueden ser divididas en dos tipos de políticas: las políticas de detección de conflictos y las políticas de solución de conflictos.

Una transacción α está en conflicto cuando su precondición no se cumple, una precondición está constituida por un conjunto de condiciones que deben cumplirse para que una transacción pueda ser ejecutada exitosamente, las precondiciones son el método por excelencia utilizado para detectar conflictos entre transacciones.

Las políticas de detección de conflictos son divididas en sintácticas y semánticas al igual que las políticas de ordenamiento de transacciones ya que están fuertemente relacionadas. En sistemas con políticas de detección de conflictos sintácticas los conflictos son detectados teniendo en cuenta la relación sucedió antes o alguna aproximación a esta relación, por ello estás políticas detectan conflictos tomando como base las marcas de tiempo aplicadas a las transacciones y generalmente detectan que dos transacciones están en conflicto cuando son ejecutadas concurrentemente y su ejecución afectan las réplicas diferentes de un mismo objeto. Las políticas sintácticas son más simples y generales, debido a esto regularmente ya vienen definidas implícitamente en los DBMS.

Las políticas semánticas usan el conocimiento de las semántica de las transacciones para detectar los conflictos, las políticas semánticas son más expresivas que las sintácticas, por ello están deben ser definidas explícitamente por el diseñador del sistema. Al utilizar políticas semánticas se pueden definir procedimientos en los cuales se indican una serie de precondiciones para los diferentes tipos de transacciones que se ejecutan en el sistema, dichos procedimientos deben ser utilizados al momento de detectar conflictos entre transacciones. Este tipo de políticas disminuye considerablemente el número de conflictos encontrados durante el proceso de reconciliación llevado a cabo en los sitios maestros.

En conclusión se puede decir que las políticas sintácticas de detección de conflictos son más simples y genéricas pero pueden ocasionar una mayor número de conflictos, en cambio, las políticas semánticas son más flexibles y complejas y deben ser definidas explícitamente por el diseñador del sistema, ocasionando un esfuerzo considerable de trabajo de ingeniería adicional.

Las políticas de solución de conflictos establecen las decisiones o acciones a tomar cada vez que se detecta un conflicto. El rol de las políticas de solución de conflictos es ejecutar o abortar las transacciones que se encuentran pendientes debido a un conflicto. Las políticas de solución de conflictos pueden definirse de manera manual o automática en los DMBS. La forma manual de solucionar conflictos es que cada vez que se detecte un conflicto el administrador de todo el sistema debe encargarse de darle solución, por lo tanto el administrador es quién toma la decisión de determinar qué se debe hacer cuando ocurre un conflicto, esta opción es poco práctica debido a que requiere la participación constante de personal humano en la solución de conflictos. La forma automática de solucionar conflictos se basa en aceptar las políticas que trae por defecto o definir las políticas propias de solución de conflictos en el DBMS, en este caso el sistema automáticamente con base en la políticas definidas tomas las decisiones de manera automática cada vez que ocurre un conflicto, normalmente es recomendable definir las políticas de solución de conflictos de manera particular en cada sistema, esto toma un tiempo considerable y en sistemas a gran escala es un trabajo complejo, sin embargo la definición de políticas propias permite preveer y estar seguros de las decisiones que tomará el DBMS cada vez que se detecta un conflicto, adicionalmente las políticas propias permiten adaptar el sistema a las diferentes restricciones impuestas por los procesos de negocio o aplicaciones que este de debe soportar.

G. Características y capacidades de los dispositivos móviles

Las características y capacidades de los dispositivos móviles es otro factor que debe ser tenido en cuenta al momento de definir los dispositivos que serán utilizados para cumplir el rol de los sitios esclavos. Las características que deben ser tenidas en cuenta en los dispositivos se centran en: la capacidad de almacenamiento secundario, ya que debe tenerse en cuenta el tamaño que ocupan los datos que desean replicarse en el dispositivo, el tamaño que ocupa la instalación del software del DBMS y las aplicaciones que serán ejecutados en el dispositivo; la capacidad en memoria principal, ya que debe verificar el dispositivo tenga memoria principal suficiente para permitir la ejecución de todos lo programas que requieren ser ejecutados para el correcto funcionamiento del sistema de replicación; el soporte de las aplicaciones que deben ejecutadas en el dispositivo, pues algunos dispositivos no soportan aplicaciones específicas a pesar de tener los recursos

físicos requeridos; y que soporte los enlaces de comunicación adecuados para que pueda ser realizado el intercambio de transacciones con los sitios maestros.

H. Características de los sistemas de los sistemas manejadores de bases de datos (DBMS)

El sistema manejador de bases de datos DBMS que se debe utilizar para implementar la tecnología es otro aspecto que debe ser tenido en cuenta al momento de diseñar un sistema que la implemente, por ello es conveniente que una vez se hayan analizado los aspectos de los numerales 4.1 a 4.6, se proceda a realizar una comparación entre los requerimientos del diseño realizado y las características que actualmente brindan los sistemas manejadores de bases de datos que existen en el mercado, como es el caso de Oracle, SQL Server, DB2, Postgres, MySQL, entre otros. La comparación realizada se considera conveniente ya que permite establecer qué requisitos del sistema que han sido diseñados pueden ser satisfechos por los DBMS actuales y cuáles requisitos deberían ser modificados. Los DBMS actualmente soportan la tecnología de replicación desconectada de bases de datos distribuidas en dispositivos móviles, sin embargo, la implementación involucra la configuración de los diferentes aspectos analizados en este artículo como son la cantidad de sitios maestros que debe tener el sistema, los esquemas de replicación utilizados, las políticas de propagación de transacciones, las políticas de ordenamiento de transacciones, las políticas de detección y solución de conflictos, entre otros. Otro aspecto muy importante que debe ser examinado es que los DBMS tengan versiones que permitan utilizar las técnicas de replicación en dispositivos móviles, por ejemplo Oracle proporciona la versión Oracle Lite y SQL Server proporciona la versión Compact Edition.

V. Conclusiones

La “replicación desconectada de bases de datos distribuidas en dispositivos móviles” es una tecnología que brinda nuevas oportunidades para mejorar algunos procesos de negocio de algunas empresas o brindar nuevos servicios en otros ámbitos, sin embargo los diseñadores de sistemas que involucran el uso de esta tecnología deben tener en cuentas varios aspectos muy importantes que le permitirá al sistema tener un adecuado funcionamiento en lo relacionado con: el control en la concurrencia, la escalabilidad, la disponibilidad de los datos, la tolerancia a fallas y las diferentes políticas y mecanismos utilizados para garantizar la consistencias de los datos.

Los principales factores que deben ser tenidos en cuenta en el diseño de un sistema que utilice la tecnología analizada son: primero, el número de sitios maestros que se requieren, por ello se debe analizar si se necesita un esquema de un único servidor maestro o de varios servidores maestros. Segundo, el modo de operación desconectado que requieren los sitios esclavos del sistema, para ello se deben analizar los permisos de lecturas y escrituras que deben ser asignados a las réplicas ubicadas en los diferentes sitios del sistema. Tercero, la topología de propagación de las transacciones, esto hace referencia a la topología física utilizada para intercambiar las transacciones ejecutadas de manera local en los diferentes sitios y que luego deben ser enviadas a los sitios maestros. Cuarto, la forma de propagar las actualizaciones, las dos principales son por transferencia de estado y por transferencia de operación, para el caso de las bases de datos se considera adecuado utilizar el esquema de propagación por transferencia de operación ya que este esquema permite reducir los conflictos ocasionados por las transacciones cuando se realiza el proceso de reconciliación y adicionalmente disminuye el tráfico de red intercambiado entres los diferentes sitios de manera considerable. Quinto, las políticas de ordenamiento de transacciones las cuales buscan establecer el orden en que deben ser ejecutadas las transacciones en todos los sitios, el principal, objetivo de estas políticas es evitar los conflictos y garantizar la consistencia de los datos Sexto, las políticas de gestión de conflictos que se encargan de definir cuándo dos transacciones están en conflicto y que decisiones se deben tomar en el sistema cuando se detecta un conflicto. Séptimo, las capacidades que tienen los dispositivos móviles en lo relacionado a almacenamiento, memoria principal, soporte para aplicaciones y enlaces de comunicación. Octavo, las capacidades y restricciones que tienen los DBMS actuales.

El diseño de sistemas que involucran el uso de la tecnología analizada a largo del artículo es un proceso que debe tener en cuenta muchos factores, por ello debe ser realizado por personal especializado en la tecnología con el fin de que se pueda obtener un correcto, adecuado y eficiente funcionamiento de los sistemas que la utilizan.

Referencias

[1] Christoph Gollmick. Replication in mobile database environments: a client-oriented approach. *Proceedings of the 14th International Workshop on Database and Expert Systems Applications IEEE*, (September 2003), pp. 1-2.

[2] Badri Nath. Transaction-Centric Reconciliation in Disconnected Client–Server Databases. *Mobile Networks and Applications*. Vol. 9, No. 5, (October 2004), pp. 459-471.

[3] José Maria Monteiro, Ângelo Brayner y Sérgio Lifschitz. A Mechanism for Replicated Data Consistency in Mobile Computing Environments. *Proceedings of the 2007 ACM symposium on Applied computing*. (March 2007), pp. 914-919..

[4] Yanli Xia, Abdelsalam (Sumi) Helal. A Dynamic Data/Currency Protocol for Mobile Database Design and Reconfiguration. *Proceedings of the 2003 ACM symposium on Applied computing*. (March 2003), pp. 550-556.

[5] Wai Gen Yee, Michael J. Donahoo, Edward Omiecinski y Shamkant B. Navathe. Scaling replica maintenance in intermittently synchronized mobile databases. *Proceedings of the tenth international conference on Information and knowledge management.* (October 2001), pp. 450-457.

[6] Abdelkrim Beloued, Jean-Marie Gilliot, Maria-Teresa Segarra, Françoise André. Dynamic data replication and consistency in mobile environments. *Proceedings of the 2nd international doctoral symposium on Middleware.* (November 2005), pp. 1-5.

[7] Takahiro Hara. Replica allocation methods in ad hoc networks with data update. *Mobile Networks and Applications*. Vol. 8, No. 4, (August 2003), pp. 343-354.

[8] David Ratner, Peter Reiher, Gerald J. Popek y Geoffrey H. Kuenning. Replication requirements in mobile environments. *Mobile Networks and Applications*. Vol. 6, No. 6, (November 2001), pp. 525-533.

[9] Yasushi Saito y Marc Shapiro. Optimistic replication. *ACM Computing Surveys (CSUR)*. Vol. 37, No. 1, (March 2005), pp. 42-81.
[10] David Ratner, Peter Reither and Gerald J. Popek. Roam: a scalable replication system for mobility. *Mobile Networks and Applications*. Vol. 9, No. 5, (October 2004), pp. 537-544.
[11] Joanne Holliday, Divyakant Agrawal y Amr El Abbadi. Disconnection modes for mobile databases. *Wireless Networks*. Vol. 8, No. 4, (Julio 2002), pp. 391-402.
[12] Subhasish Mazumdar y Panos K. Chrysanthis. Localization of integrity constraints in mobile databases and specification in PRO-MOTION. *Mobile Networks and Applications*. Vol. 9, No. 5, (October 2004), pp. 481-490.